

## CSE575-Section4: MyLocation activity

- I. Giving access to the location service (GPS) of Android device by adding this line of code in to the AndroidManifest.xml

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

It should look like this AndroidManifest.xml



- II. Layout activity\_my\_location.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tando.school.MyLocation"
    android:orientation="vertical">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="15dp"
                android:layout_marginBottom="20dp"
                android:gravity="center"
                android:text="Current Location"
                android:textColor="#000"
                android:textSize="20dp" />

            <TextView
                android:id="@+id/txtLat"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:layout_marginTop="10dp"
                android:text="Latitude: "
                android:textColor="#000"
                android:textSize="15dp" />

            <TextView
                android:id="@+id/txtLong"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="15dp"
                android:layout_marginTop="10dp"
```

```

        android:text="Longitude: "
        android:textColor="#000"
        android:textSize="15dp" />

<TextView
    android:id="@+id/txtAccuracy"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="10dp"
    android:text="Accuracy: "
    android:textColor="#000"
    android:textSize="15dp" />

<TextView
    android:id="@+id/txtAltitude"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="10dp"
    android:text="Altitude: "
    android:textColor="#000"
    android:textSize="15dp" />

<TextView
    android:id="@+id/txtAddress"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="10dp"
    android:text="Address: "
    android:textColor="#000"
    android:textSize="15dp" />
</LinearLayout>
</ScrollView>
</LinearLayout>

```

### III. MyLocation.java (Explanations are in comments)

```

package com.tando.school;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Build;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;

import java.io.IOException;
import java.util.List;
import java.util.Locale;
//Must give permission ACCESS_FINE_LOCATION in the manifest

```

```

public class MyLocation extends AppCompatActivity {
    // provides access to the system location services
    LocationManager locationManager;
    //Used for receiving notifications from the LocationManager when the location
    has changed
    LocationListener locationListener;
    //Process when users give the permission
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        //Start the location service when users allow for the location service
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            startListening();
        }

        //checking method for granted permission
        public void startListening() {
            if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
                locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
            }
        }
        //Update location method. When the user moves to another location
        public void updatedLocationInfo(Location location) {
            //testing on log
            Log.i ("Location", location.toString());
            //Cast into their IDs textviews to parse information from location service
            TextView latTextView = (TextView) findViewById(R.id.txtLat);
            TextView lonTextView = (TextView) findViewById(R.id.txtLong);
            TextView altTextView = (TextView) findViewById(R.id.txtAltitude);
            TextView accTextView = (TextView) findViewById(R.id.txtAccuracy);
            //Retrieve latitude, longitude, altitude, accuracy from the location
            service
            latTextView.setText("Latitude: " + location.getLatitude());
            lonTextView.setText("Longitude: " + location.getLongitude());
            altTextView.setText("Altitude: " + location.getAltitude());
            accTextView.setText("Accuracy: " + location.getAccuracy());

            /**
             * Create Geocoder object to Get the address
             * A class for handling geocoding and reverse geocoding. Geocoding is the
            process of
             * transforming a street address or other description of a location
             * into a (latitude, longitude) coordinate.
             * reference:
            https://developer.android.com/reference/android/location/Geocoder.html
             */
            Geocoder geocoder = new Geocoder(getApplicationContext(),
Locale.getDefault());
            //Using try/catch to prevent the app from crashing when failing to get
            Addresses from Geocoder
            try {
                //Declare the error string
                String address = "Unable to get the address!";
                //Returns an array of Addresses that are known to describe the area
            immediately surrounding the given latitude and longitude, return only 1 address
                List<Address> listAddresses =
geocoder.getFromLocation(location.getLatitude(), location.getLongitude(), 1);
                if (listAddresses != null && listAddresses.size() > 0) {

```

```

        //Set address to empty string again when we know it is working
        address = "Address: \n";
        //check for every item in the list Addresses is valid
        if (listAddresses.get(0).getSubThoroughfare() != null) {
            address += listAddresses.get(0).getSubThoroughfare() + " ";
        }
        //Street name
        if (listAddresses.get(0).getThoroughfare() != null) {
            address += listAddresses.get(0).getThoroughfare() + "\n";
        }
        //City name
        if (listAddresses.get(0).getLocality() != null) {
            address += listAddresses.get(0).getLocality() + "\n";
        }
        //Zip code
        if (listAddresses.get(0).getPostalCode() != null) {
            address += listAddresses.get(0).getPostalCode() + "\n";
        }
        //Country name
        if (listAddresses.get(0).getCountryName() != null) {
            address += listAddresses.get(0).getCountryName() + "\n";
        }
    }

    TextView addressTextView = (TextView) findViewById(R.id.txtAddress);
    //set the array of address into the text View declared above
    addressTextView.setText(address);

} catch (IOException e) {
    e.printStackTrace();
}
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_my_location);
    locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
    //Retrieve the new address list from locationManager when the users change
    their location
    locationListener = new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {
            //Call the updatedLocationInfo method.
            updatedLocationInfo(location);
        }
        //These method will not be used
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras)
    {

    }

    @Override
    public void onProviderEnabled(String provider) {

    }

    @Override
    public void onProviderDisabled(String provider) {

    }
}

```

```

};
//Check for the version of SDK
if (Build.VERSION.SDK_INT < 23) {

    startListening();
} else {
    // above 23 we need to check for permission
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        //ask for permission. Number 1 is just a request queue.
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
    }
    //we have permission
    else {

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
locationListener);
        //Returns a Location indicating the data from the last known
location fix obtained from the given provider
        Location location =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
        //In case location does not have lastknownlocation, we call the
updatedLocation method above
        if (location != null) {
            updatedLocationInfo(location);
        }
    }
}
}
}

```

Demo:

1. When click on it as the first time, it will ask for a permission



1:01



## My Location

### Current Location

Latitude:

Longitude:

Accuracy:

Altitude:

Address:



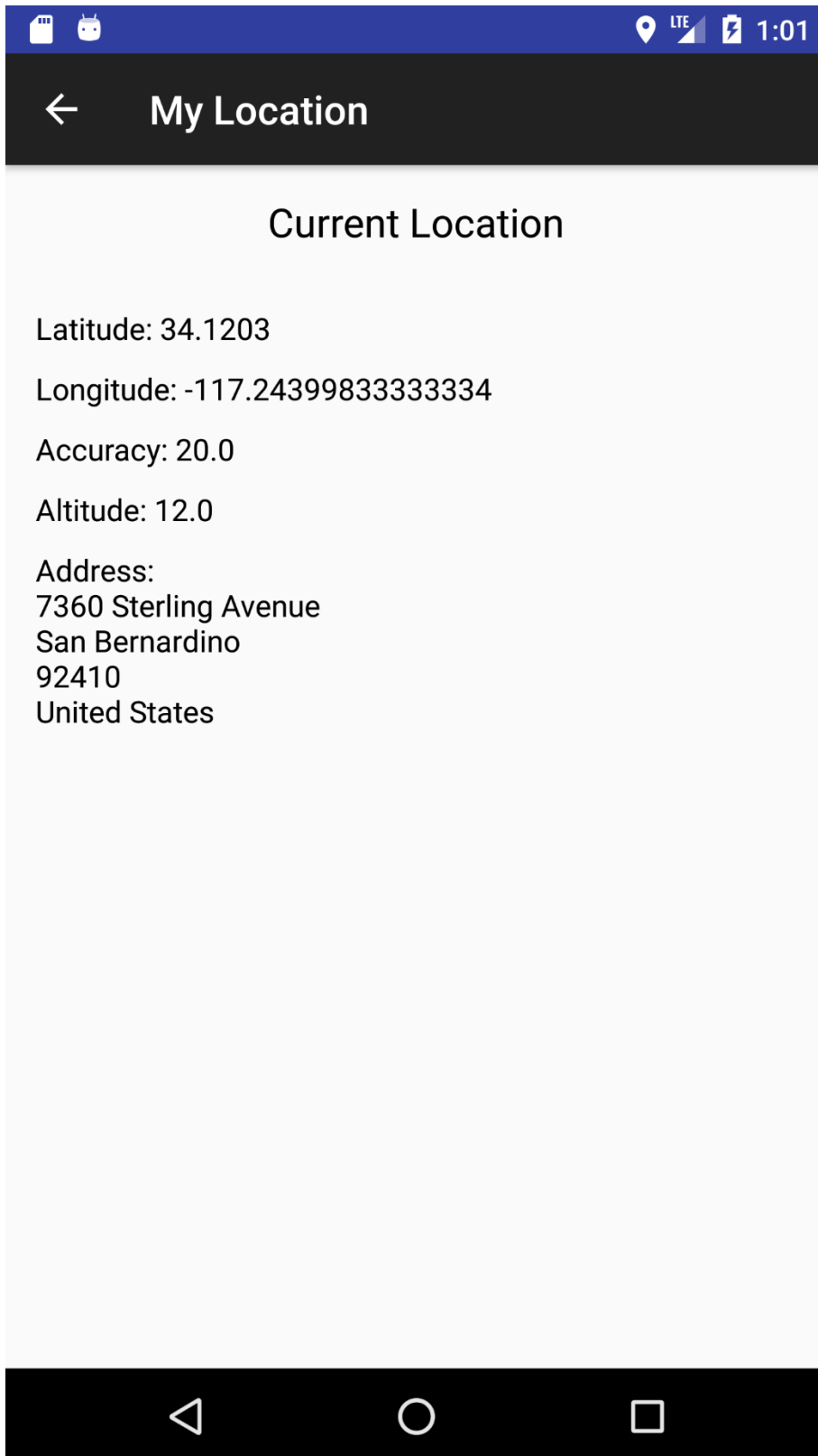
Allow **School** to access  
this device's location?

DENY

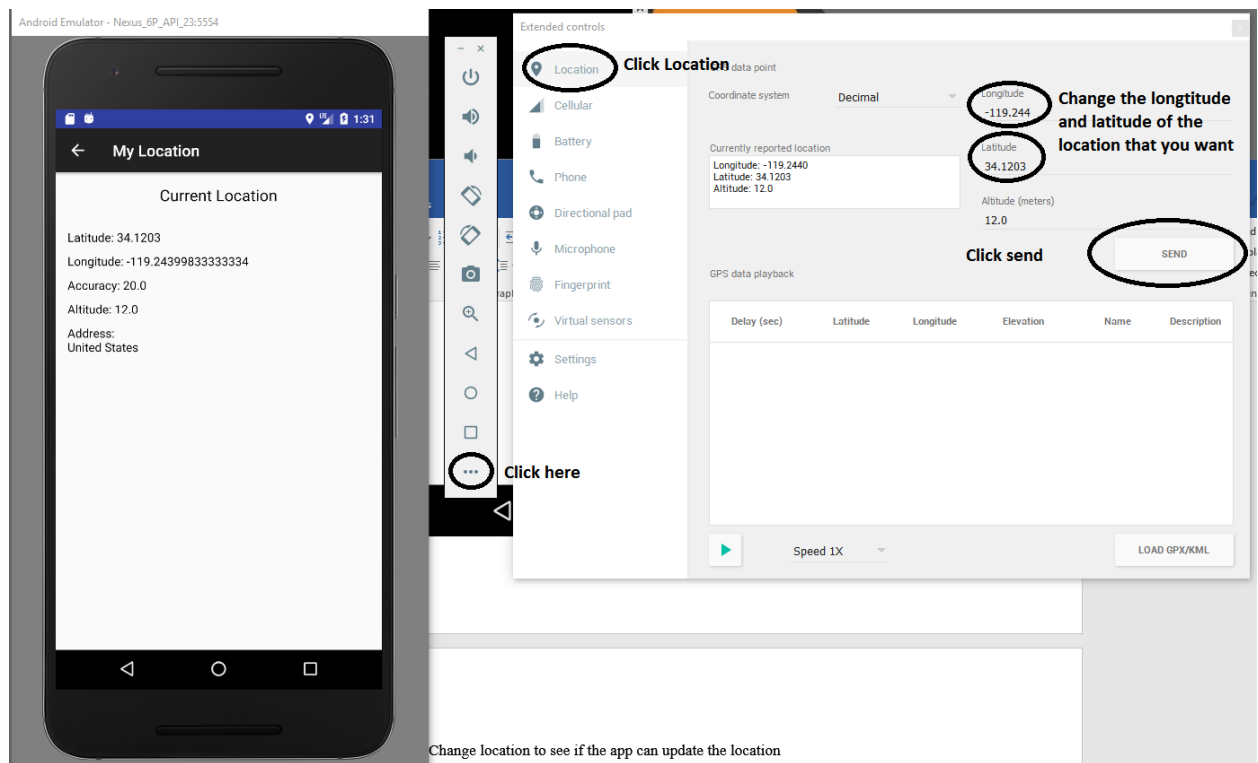
ALLOW



2. Click allow, you will not see anything display, click the back arrow and click on the My Location section again. You will see your current location display



### 3. Change location to test if the app can update the location



### 4. The location will change when we click send (In this case I change from my house to CSUSB campus)

